Roma Tre

Relazione Ingegneria dei Dati

Piergiorgio Fornaro (577925)

03/11/2025

INDEXER E SEARCHER SU LUCENE

---

1 Introduzione

La seguente relazione ha lo scopo di rendicontare i risultati ottenuti in seguito alla creazione di codice per sfruttare Lucene come Indexer e Searcher su un piccolo database di dati.
La versione di Lucene utilizzata è la 10.3.1, con JDK 25.0.1.
Sono stati analizzati due campi principali per ogni documento: **Nome** e **Contenuto** ed è stata realizzata una GUI per poter interagire in maniera più agevole con il motore di ricerca.

1.1 GitHub del progetto

URL: https://github.com/PiergiF/IngegneriaDeiDati_25-26_LuceneHomework2

2 Atricolazione del progetto

Nella sua prima versione il progetto è scritto tutto in una classe per verificarne l'effettivo funzionamento. In futuro verrà Rifattorizzata in più classi e con l'aggiunta di test automatici e non manuali.
Il codice è così articolato:

- Prima parte contenente la **creazione** e l'inizializzazione della **GUI**, con tutti i pulsanti e la dark mode.
- Possibilità di **scelta della directory** da dove prendere i dati da indicizzare (ovviamente è presente una directory di default).
- Parte di **indicizzazione** in cui vengono visti i file nella directory dei dati selezionata. Per ogni file legge ed indicizza:
  - **Nome**: ovvero il nome del file. L'obiettivo della ricerca su questo campo è la corrispondenza esatta ma case insensitive. A tale scopo è stato utilizzato un TextField.
  - **Contenuto**: ovvero ciò che è scritto dentro al file. L'obiettivo della ricerca su questo campo è trovare le parole cercate o che contengono quella corrispettiva stringa al loro interno. Quest'ultima cosa è resa possibile grazie all'aggiunta del carattere speciale * indicante "è presente qualsiasi cosa in questo punto". É stato utilizzato l'ItalianAnalyzer di Lucene, che gestisce la lingua italiana, rimuovendo le stopwords, uniformando i caratteri ed effettuando lo stemming (riduzione di una parola alla sua forma radice) per una ricerca più robusta.

Se non è stato trovato nessun file, il problema viene segnalato all'utente. Si utilizza un IndexWriter per scrivere i documenti nell'indice. A fine indicizzazione viene riportato il tempo totale e, se selezionato, anche il tempo impiegato per ogni file. É poi possibile salvare i risultati in formato *csv*. Si può ricreare l'inidice direttamente da programma, tramite apposito tasto che, se premuto, cancellerà il vecchio indice, creandone uno nuovo.

- Parte di **Ricerca**, che si occupa della ricerca dei documenti indicizzati. Apre l'indice salvato su disco e utilizza un IndexSearcher per interrogare i documenti. In base al campo selezionato sa su quale campo cercare e costruisce la Query tramite un QueryParser. La query ritorna ciò che si è cercato, il numero di occorrenze, l'eventuale file di riferimento (nel caso sia un contenuto) e lo score.
- Parte di **Esportazione** dei risultati ottenuti.
- Parte dedicata al **Log**.
- **Main** per far partire l'applicazione.

3 Test

I test sono stati momentaneamente realizzati a mano.
Sono state effettuate query su 3 file *txt*:

- Roma.txt
- Lucene.txt
- Esercizi.txt

Query realizzate:

- "Roma.txt" con ricerca sul nome. Risultato: Ricerca [nome]Roma.txt - Risultati: 1 - Roma.txt (score: 0,446)
- "Roma.json" con ricerca sul nome. Ricerca [nome]: Roma.json - Risultati: 0
- "Roma" con ricerca sul nome. Ricerca [nome]: Roma - Risultati: 0
- "ROMA.TXT" con ricerca sul nome. Ricerca [nome]: ROMA.TXT - Risultati: 1 - Roma.txt (score: 0,446)
- "Software" con ricerca sul contenuto. Ricerca [contenuto]: Software - Risultati: 1 - lucene.txt (score: 0,455)
- "open source" con ricerca sul contenuto. Ricerca [contenuto]: open source - Risultati: 1 - lucene.txt (score: 0,593)
- "opensource" con ricerca sul contenuto. Ricerca [contenuto]: opensource - Risultati: 0
- "finire tutti gli esercizi" con ricerca sul contenuto. Ricerca [contenuto]: finire tutti gli esercizi - Risultati: 1 - esercizi.txt (score: 1,215)
- "esa*" con ricerca sul contenuto. Ricerca [contenuto]: esa* - Risultati: 1 - esercizi.txt (score: 1,000)
- "Paulo Dybala" con ricerca sul contenuto. Ricerca [contenuto]: Paulo Dybala - Risultati: 1 - Roma.txt (score: 1,170)

3.1 Tempi di indicizzazione

Sono stati indicizzati 3 file con i seguenti tempi:

- Indicizzato: lucene.txt (2,1 ms)
- Indicizzato: Roma.txt (0,3 ms)
- Indicizzato: esercizi.txt (0,3 ms)

Indicizzazione completata in 0,025 secondi (3 file, 118,71 file/sec)

4 Sviluppi futuri

In futuro gli obiettivi sono di:

- Rifattorizzare in più classi.
- Aggiungere test automatici.
- Aggiornare la relazione.
- Aggiungere nuove feature.